Technical Disclosure Commons

Defensive Publications Series

November 2020

ROOT CAUSE IDENTIFICATION

John Garrett

Fang Chen

Adrian Bjune

ROOT CAUSE IDENTIFICATION

AUTHORS:
John Garrett
Fang Chen
Adrian Bjune

ABSTRACT

Presented herein is a novel machine learning approach that learns the failure patterns of a device and uses this information to drive automatic root case identification. Anomalous events can be associated with potential root causes in order to build intellectual capital for training future predictive automated remediation systems.

DETAILED DESCRIPTION

Network elements can produce hundreds of possible telemetry streams. When an event occurs that impacts a network element, leading indicators to the event are likely in just a few of the streams. Effects from the event are also in the streams. When operators examine the elements after the fact, understanding which streams are associated with the cause and which streams are a result of the event is not easy. Further, extracting only the useful information for troubleshooting (now) and building predictive models (later) can be a challenge with the possible volume of data available.

This proposal provides a novel approach for identifying root cause telemetry streams and patterns within hundreds of possible streams in order to build a data set for predictive, real time automated remediation and risk avoidance.

The solution begins with a set of models that can be fitted and run remotely (on a device or within a Local Area Network (LAN)) to evaluate each telemetry stream to generate a single evaluation metric. This metric could be any function that produces a single value, such as a standard score (also known as a 'Z-Score'), distance from predicted or anomalous level, etc.

During operation, each device can produce many telemetry streams. Once the evaluation metric for each single telemetry stream is obtained, this metric is used to represent the stream in the next level. There are a finite number of representations. Consider,

for example, three representations, identified in red, yellow, and blue, as shown below in Figure 1.

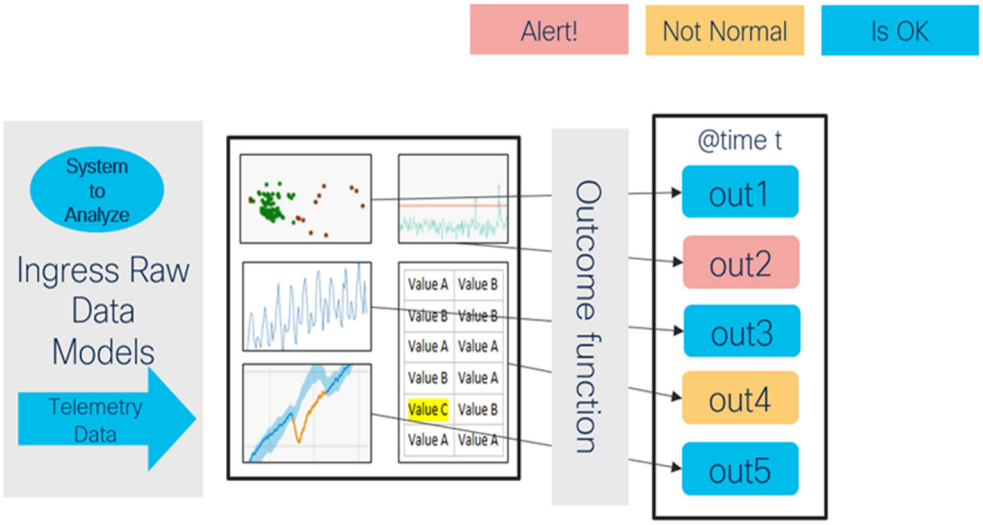

Figure 1: Data Value Streams

For each time period, only the evaluated output is retained to represent the stream at that point in time (@time, t). These evaluations become a new time series. This new data set is evaluated by sliding a simple straddling convolutional neural network (CNN) layer or, more generally, a convolution layer across the time slices, as shown below in Figure 2.

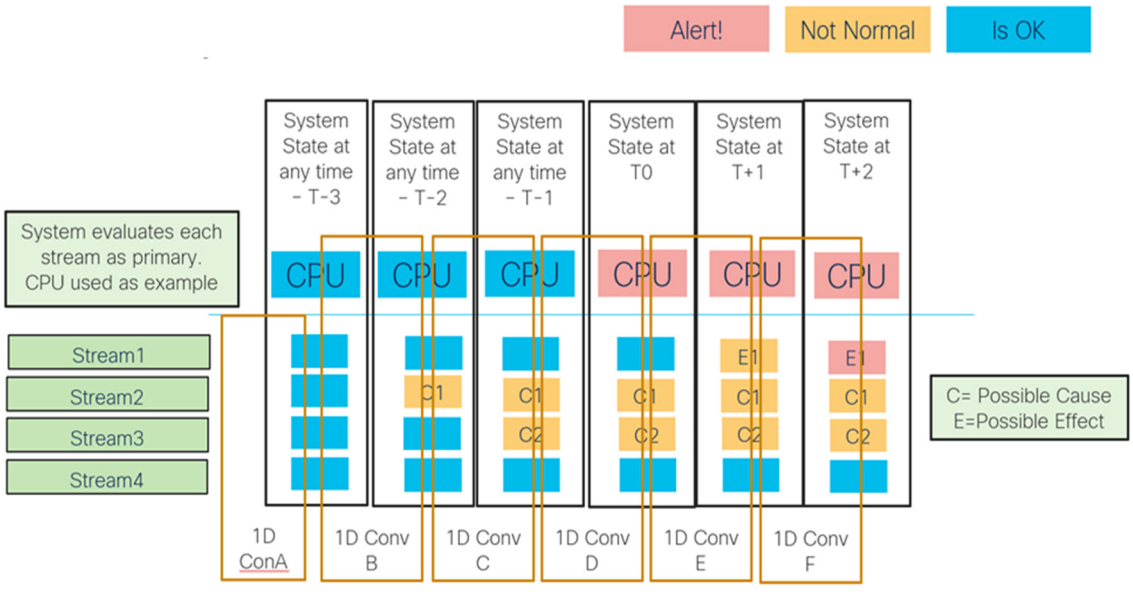

Figure 2: Second 'Convolution' Layer Setup

When the dimensionality is high, assuming there are many telemetry streams, the convolution layer can also be used to extract the key features from the streams. In this case,

a new, smaller data set of 'state transitions' is produced. This new data set, also referred to herein as a 'convolution set', will be much smaller and only contain streams that made a state transition from one time period to another. Figure 3, below, illustrates example details associated with the convolution set.

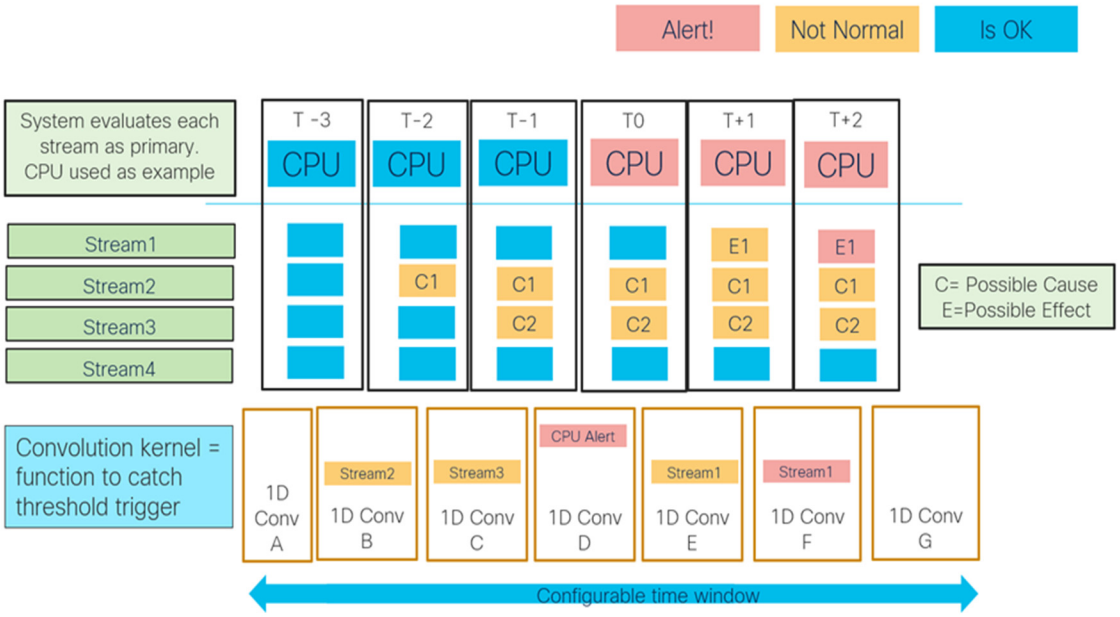

Figure 3: Convolution Set

The purpose of this new layer is to develop a sequential pattern of streams that changed state around any single event of interest. Any telemetry stream that has a trigger (e.g., Anomaly, threshold) could be the 'single event of interest'. Due to the time nature of the analysis, it can be inferred that streams that changed state before the alert are possible causes and streams that changed state after the event are effects. The entire set of state transitions that can be bookended by periods of non-transitions (or multiple periods if desired) is considered a single 'event'. In various implementations, the length of the configuration time window can be set to 1-day, 3-day, and 7-day values to account for short, mid, and long-term effects, respectively. Figure 4, below, illustrates example details associated with developing a sequential pattern of streams.

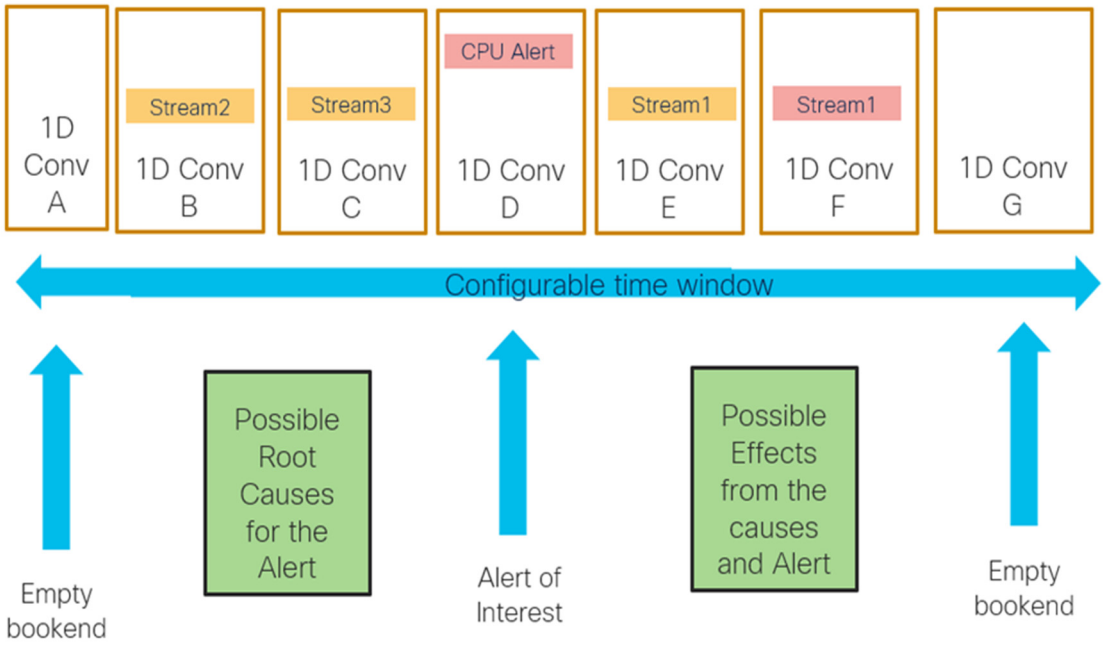

Figure 4: Sequential Pattern of Streams

This identification of possible root causes and their source data is valuable for determining the exact cause of the event of interest. The troubleshooting can be targeted at these items, by bringing in the raw telemetry data, using reasoner approaches, and also checking for any known alerts that may be related (critical bug matching, security advisories, etc.). Each sequential pattern of changes over N time periods is captured as a 'transaction' or single event, which can be used for training sequential pattern mining algorithms that can recognize events happening in real time.

Consider a simple scenario involving a significant change in Media Access Control (MAC) address rewrites, a significant increase in traffic counters through some interfaces, followed by some level of resource exhaustion (e.g., processor, available bandwidth, etc.), followed by many other processes alarming (say processes x, y, z). In this example, the sequential pattern can be assembled from the state changes of all of these factors and any target event (say y) can have multiple sequential patterns that lead to it. Each pattern is a new kind of knowledge – a record that can be used for predictive Sequential Pattern Matching (SPM) models as illustrated in Figure 5.

Finally, a user feedback function can be used to identify if the root causes were correct, as illustrated in Figure 5, below. If so, the remediation activities are collected for

future automated remediation, and the record is stored to be used for predictive use cases where automated could trigger within the time window to prevent the issue from happening. A positive user feedback is used to capture a useful record for sequential pattern mining, as well as any metadata that may be useful for automated remediation efforts. Monitoring the effects returning to normal operation can be a machine 'confirmation' that the record is a known pattern that was corrected (once automated remediation is in place).

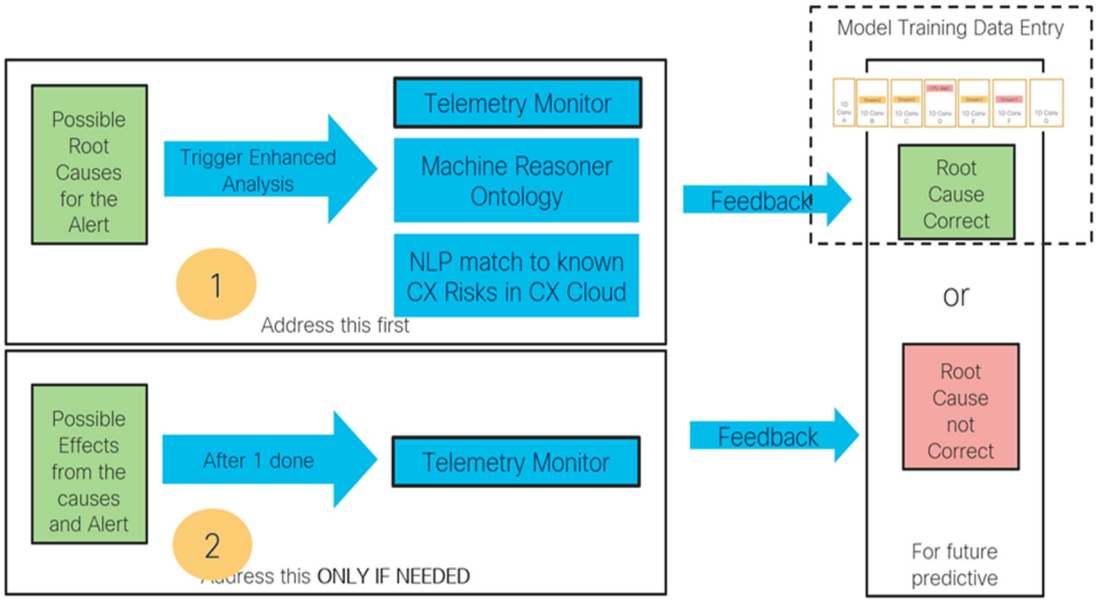

Figure 5: Collecting User Feedback to Identify if Root Causes are Correct

If root causes are not correct, this can also be reflected, and system could gather what data streams should have been leading indicators in order to improve the remote models. In this case, the system is capable of learning from both positive and negative feedback and can improve accordingly.

The root case association/identification is a proactive approach. Figure 6(a) below illustrates the existing workflow for anomaly detection. First, the system identify anomalies, but it does not have capability of automatic root cause identification, therefore, it can take extensive time to identify issues, which potentially may lead to system down time. Once an expert fixes the issue, the problem is resolved and network up and running again.

Figure 6(a): Reactive Anomaly Detection Roadmap

Figure 6(b): Proactive Anomaly Detection Roadmap

In contrast, Figure 6(b) illustrates the proposed predictive system for root cause identification. As illustrated in Figure 6(b), once an anomaly is observed, a list of potential root cause(s) will be provided to network engineers to assist decision-making, or to automated remediation or reasoning system. Then network engineers or the automated systems can place preventive measures or remediation accordingly as quickly as possible. As a result, the network will continue to operate without a down period.

Thus, a goal of the solution presented herein is to identify preceding alarms, the parent in alert state (assume ANY stream can be the "parent"), and then child alarms that are a result of both the preceding and parent events. Further, after user confirmation, the entire sequence of state changes is a sequential pattern to be used with sequential pattern algorithms for real time predictions of the parent events and the effects they will have. Accordingly, this solution provides a novel method for capturing a sequential pattern of state changes for building future predictive models.

Further, this solution is unique compared to classic causality inference tests, such as Granger causality or Sims causality. Consider Granger causality as one example. For example consider two time series streams: {x1, x2, . . ., xT} and {y1, y2, . . ., yT} in which x1, x2, . . ., xT-1 can be used to predict 'xT' and an error 'e1'. Both information from 'x' and 'y' can be used to predict xT, essentially using {x1, x2, . . ., xT-1, y1, y2, . . ., yT-1} to predict xT and got an error e2. Now, if it is observed that e2< e1, then the joint feature of x and y is better at prediction xT than just using x feature alone, therefore y is effective for predicting x, so y and x have a causal relationship. This approach is straightforward but it

suffers from some drawbacks when compared with our machine learning approach of the solution provided herein.

First, there are curses of dimensionality. The above example only has two dimensions. However, if the feature space is of a high dimension (which it will as there are many telemetry streams coming from each single device), then modeling and testing will be needed for n! (factorial of n) time considering every single combination. Secondly, the modeling approaches used in these classic approaches are mostly straight forward (linear functions or moment functions) which are not representative enough for cases discussed herein.

For the machine learning (ML) approach described herein, the strength of the ML model can be leveraged and does not make any assumptions of the relationships between features and outputs. More importantly, the convolutional neural network (CNN) layer of this proposal is used as a middle layer to extract essential information from the original feature space, therefore the causality can be derived using these embedded essential information without having to explore all n! solution spaces.

In summary, a novel machine learning approach is provide herein that learns the failure patterns of a device and uses this information to drive automatic root case identification. Anomalous events can be associated with potential root causes in order to build intellectual capital for training future predictive automated remediation systems.